

CU-AssignmPrint L xRemove DReverse LiDelete thMerge TwRemove DLinked ListReverse LiRotate ListSort ListLinked List+

<>Cgeeksforgeeks.org/problems/print-linked-list-elements/0

AgodaAmazonAcer(4186)YouTubeAll Bookmarks90% RefundCoursesTutorialsJobsPracticeContests

ProblemEditorialSubmissionsComments

My SubmissionsAll Submissions

Refresh

Time (IST)	Status	Marks	Lang	Test Cases	Code
2025-02-14 16:49:04	Correct	0 ?	cpp	1112/1112	View
2025-02-14 16:42:21	Wrong	0	cpp	0/1112	View
2025-02-14 16:40:29	Wrong	0	cpp	0/1112	View
2025-02-14 16:38:33	Wrong	0	cpp	0/1112	View

C++ (g++ 5.4)

Start Timer

1. // ...code ends  
19. /\*  
20. struct Node {  
21. int data;  
22. struct Node\* next;  
23. };  
24. Node(int x) {  
25. data = x;  
26. next = nullptr;  
27. }  
28. };  
29. \*/  
30. /\*  
31. Print elements of a Linked List on console  
32. Head pointer input could be NULL as well for empty list  
33. \*/  
34.   
35. class Solution {  
36. public:  
37. // Function to display the elements of a Linked List in same line  
38. void printlist(Node \*head) {  
39. // your code goes here  
40. while (head != NULL)  
41. {  
42. cout << head->data;  
43. if (head->next) cout << " ";  
44. head = head->next;  
45. }  
46. }  
47. };  
48. // ...code ends

Custom InputCompile & RunSubmit

Upcoming Earnings

Search

ENG IN16:5614-02-2025

CU-Assign | Print Link | Remove | Reverse Li | Delete the | Merge Two | Remove D | Linked List | Reverse Li | Rotate List | Sort List | Linked List | + | - | X

leetcode.com/problems/remove-duplicates-from-sorted-list/submissions/ | Agoda | Amazon | Acer | (4186) YouTube | All Bookmarks

Problem List | Run | Submit | Premium

Description | Editorial | Solutions | Submissions

Status	Language	Runtime	Memory	Notes
1 Accepted Feb 05, 2025	C++	0 ms	16.1 MB	

Code

```
13 ListNode* deleteDuplicates(ListNode* head) {
14     ListNode* res = head;
15
16     while (head && head->next) {
17         if (head->val == head->next->val) {
18             head->next = head->next->next;
19         } else {
20             head = head->next;
21         }
22     }
23
24     return res;
25 }
26 ;;
```

Saved | Ln 1, Col 1

Testcase | Test Result

Case 1 | Case 2 | +

</> Source

20°C  
Sunny

Search

ENG  
IN

16:53  
14-02-2025

CU-Assignm...Print Link...Remove D...Revers...Delete the...Merge Two...Remove D...Linked List...Reverse Li...Rotate List...Sort List...Linked List...+...-...x

leetcode.com/problems/reverse-linked-list/submissions/

AgodaAmazonAcer(4196) YouTubeAll Bookmarks

Problem List<>RunSubmit

DescriptionEditorialSolutionsSubmissions

Status	Language	Runtime	Memory	Notes
2Accepted Feb 12, 2025	C++	0 ms	13.3 MB	
1Accepted Feb 05, 2025	C++	0 ms	13.3 MB	

Code

C++Auto

```
1 /**
2  * Definition for singly-linked list.
3  * struct ListNode {
4  *     int val;
5  *     ListNode *next;
6  *     ListNode() : val(0), next(nullptr) {}
7  *     ListNode(int x) : val(x), next(nullptr) {}
8  *     ListNode(int x, ListNode *next) : val(x), next(next) {}
9  * };
10 */
11 class Solution {
12 public:
13     ListNode* reverseList(ListNode* head) {
14         if(head==NULL||head->next==NULL){
15             return head;
16         }
17         ListNode* current=head;
18         ListNode* prev=NULL;
```

SavedLn 1, Col 1

TestcaseTest Result

Case 1Case 2Case 3+

</> Source

20°C  
Sunny

Search

16:53  
14-02-2025

CU-Assign...Print Link...Remove D...Reverse Li...Delete x...Merge Two...Remove D...Linked List...Reverse Li...Rotate List...Sort List...Linked List...+...-...x

leetcode.com/problems/delete-the-middle-node-of-a-linked-list/submissions/AgodaAmazonAcer(4196)YouTubeAll Bookmarks

Problem List<>RunSubmitRun Submit

DescriptionEditorialSolutionsSubmissions

Status

Language

Runtime

Memory

Notes

1

Accepted  
Feb 05, 2025

C++

0 ms

312 MB

Code

C++Auto

```
1 /**
2  * Definition for singly-linked list.
3  * struct ListNode {
4  *     int val;
5  *     ListNode *next;
6  *     ListNode() : val(0), next(nullptr) {}
7  *     ListNode(int x) : val(x), next(nullptr) {}
8  *     ListNode(int x, ListNode *next) : val(x), next(next) {}
9  * };
10 */
11 class Solution {
12 public:
13     ListNode* deleteMiddle(ListNode* head) {
14         if(!head || !head->next) return nullptr;
15         ListNode*slow = head;
16         ListNode*fast = head->next->next;
17         while(fast !=nullptr && fast->next != nullptr){
18             fast = fast->next->next;
19         }
20     }
21 };
22 
```

SavedLn 1, Col 1

TestcaseTest Result

Case 1Case 2Case 3+

</> Source

20°C  
Sunny

Search

ENG  
IN

16:53  
14-02-2025

CU-Assignm...Print Link...Remove D...Reverse Li...Delete the...Merge: x...Remove D...Linked List...Reverse Li...Rotate List...Sort List...Linked List...+...-...x

leetcode.com/problems/merge-two-sorted-lists/submissions/

AgodaAmazonAcer(4196) YouTubeAll Bookmarks

Problem List<>↔

RunSubmit🕒📄

800🔧🔥0🔴Premium

DescriptionEditorialSolutionsSubmissions

Status ▾

Language ▾

Runtime

Memory

Notes

⚙️

1

Accepted

Feb 05, 2025

C++

0 ms

19.6 MB

Code

C++ ▾Auto

Ln 1, Col 1

1 /\*\*  
2 \* Definition for singly-linked list.  
3 \* struct ListNode {  
4 \* int val;  
5 \* ListNode \*next;  
6 \* ListNode() : val(0), next(nullptr) {}  
7 \* ListNode(int x) : val(x), next(nullptr) {}  
8 \* ListNode(int x, ListNode \*next) : val(x), next(next) {}  
9 \* };  
10 \*/  
11 class Solution {  
12 public:  
13 ListNode\* mergeTwoLists(ListNode\* list1, ListNode\* list2) {  
14 ListNode\* dummy = new ListNode(0);  
15 ListNode\* cur = dummy;  
16  
17 while (list1 && list2) {  
18 if (list1->val < list2->val) {

Testcase>Test Result

Case 1Case 2Case 3+

</> Source🔍

20°C  
Sunny

Search

📄🔍📁🌐🔒🗨️

ENG  
IN

16:53  
14-02-2025

CU-AssignPrint LinkRemove DReverse LiDelete thMerge TwRemo xLinked ListReverse LiRotate ListSort ListLinked List+

<>↺

leetcode.com/problems/remove-duplicates-from-sorted-list-ii/submissions/AgodaAmazonAcer(4196) YouTubeAll Bookmarks

Problem List<>⚙️RunSubmit🕒📄Premium

DescriptionEditorialSolutionsSubmissions

StatusLanguageRuntimeMemoryNotes⚙️

1AcceptedFeb 05, 2025C++0 ms15.6 MB

CodeC++Auto

```
1 /**
2  * Definition for singly-linked list.
3  * struct ListNode {
4  *     int val;
5  *     ListNode *next;
6  *     ListNode() : val(0), next(nullptr) {}
7  *     ListNode(int x) : val(x), next(nullptr) {}
8  *     ListNode(int x, ListNode *next) : val(x), next(next) {}
9  * };
10 */
11 class Solution {
12 public:
13     ListNode* deleteDuplicates(ListNode* head) {
14         ListNode dummy(0, head);
15         ListNode* prev = &dummy;
16
17         while (head) {
18             if (head->next && head->val == head->next->val) {
```

SavedLn 1, Col 1

TestcaseTest Result

Case 1Case 2+

</> Source🕒

20°C Sunny

Search

ENG IN

16:53 14-02-2025

CU-AssignPrint LinkRemove DReverse LiDelete thMerge TwRemove DLinked xReverse LiRotate ListSort ListLinked List

leetcode.com/problems/linked-list-cycle/submissions/AgodaAmazonAcer(4196)YouTubeAll Bookmarks

Problem ListRunSubmitPremium

DescriptionEditorialSolutionsSubmissions

Status	Language	Runtime	Memory	Notes
1AcceptedFeb 05, 2025	C++	8 ms	12 MB	

Code

```
1 /**
2  * Definition for singly-linked list.
3  * struct ListNode {
4  *     int val;
5  *     ListNode *next;
6  *     ListNode(int x) : val(x), next(NULL) {}
7  * };
8  */
9 class Solution {
10 public:
11     bool hasCycle(ListNode *head) {
12         ListNode* fast = head;
13         ListNode* slow = head;
14
15         while (fast != nullptr && fast->next != nullptr) {
16             fast = fast->next->next;
17             slow = slow->next;
18         }
19     }
20 }
```

SavedLn 1, Col 1

TestcaseTest Result

Case 1Case 2Case 3+

</> Source

20°C Sunny

Search

ENG IN16:5314-02-2025

CU-Assign | Print Link | Remove D | Reverse Li | Delete the | Merge Two | Remove D | Linked List | Revers | Rotate List | Sort List | Linked List | + | - | X

leetcode.com/problems/reverse-linked-list-ii/submissions/

Agoda | Amazon | Acer | (4196) YouTube | All Bookmarks

Problem List | Run | Submit | Premium

Description | Editorial | Solutions | Submissions

Status

Language

Runtime

Memory

Notes

1

Accepted  
Feb 05, 2025

C++

0 ms

11.3 MB

Code

C++ | Auto

```
1 /**
2  * Definition for singly-linked list.
3  * struct ListNode {
4  *     int val;
5  *     ListNode *next;
6  *     ListNode() : val(0), next(nullptr) {}
7  *     ListNode(int x) : val(x), next(nullptr) {}
8  *     ListNode(int x, ListNode *next) : val(x), next(next) {}
9  * };
10 */
11 class Solution {
12 public:
13     ListNode* reverseBetween(ListNode* head, int left, int right) {
14         if (!head || left == right) {
15             return head;
16         }
17
18         ListNode* dummy = new ListNode(0);
```

Saved | Ln 1, Col 1

Testcase | Test Result

Case 1 | Case 2 | +

</> Source

20°C  
Sunny

Search

ENG  
IN

16:53  
14-02-2025



CU-AssignPrint LinkRemove DReverse LDelete thMerge TwRemove DLinked ListReverse LRotate: XSort ListLinked List+All Bookmarks

leetcode.com/problems/rotate-list/submissions/

Problem ListRunSubmitPremium

DescriptionEditorialSolutionsSubmissions

Status

Language

Runtime

Memory

Notes

1

Accepted  
Feb 05, 2025

C++

0 ms

16.3 MB

Code

C++Auto

```
1 /**
2  * Definition for singly-linked list.
3  * struct ListNode {
4  *     int val;
5  *     ListNode *next;
6  *     ListNode() : val(0), next(nullptr) {}
7  *     ListNode(int x) : val(x), next(nullptr) {}
8  *     ListNode(int x, ListNode *next) : val(x), next(next) {}
9  * };
10 */
11 class Solution {
12 public:
13     ListNode* rotateRight(ListNode* head, int k) {
14         if (!head || !head->next || k == 0) return head;
15
16         ListNode* tail = head;
17         int length = 1;
18     }
19 }
```

SavedLn 1, Col 1

TestcaseTest Result

Case 1Case 2+

</> Source

20°C  
Sunny

Search

ENG  
IN

16:53  
14-02-2025

CU-AssignPrint LinkRemove DReverse LiDelete thMerge TwRemove DLinked ListReverse LiRotate ListSort LLinked List

leetcode.com/problems/sort-list/submissions/AgodaAmazonAcer(4196) YouTubeAll Bookmarks

Problem ListRunSubmit

DescriptionEditorialSolutionsSubmissions

Status

Language

Runtime

Memory

Notes

1

Accepted  
Feb 05, 2025

C++

51 ms

75.8 MB

Code

C++Auto

```
1 /**
2  * Definition for singly-linked list.
3  * struct ListNode {
4  *     int val;
5  *     ListNode *next;
6  *     ListNode() : val(0), next(nullptr) {}
7  *     ListNode(int x) : val(x), next(nullptr) {}
8  *     ListNode(int x, ListNode *next) : val(x), next(next) {}
9  * };
10 */
11 class Solution {
12 public:
13     ListNode* getmid(ListNode* head) {
14         ListNode* slow = head;
15         ListNode* fast = head->next;
16
17         while (fast != NULL && fast->next != NULL) {
18             slow = slow->next;
```

SavedLn 1, Col 1

TestcaseTest Result

Case 1Case 2Case 3

</> Source

20°C  
Sunny

Search

ENG  
IN

16:53  
14-02-2025

CU-Assign | Print Link | Remove D | Reverse Li | Delete the | Merge Two | Remove D | Linked List | Reverse Li | Rotate List | Sort List | Linked x

leetcode.com/problems/linked-list-cycle-ii/submissions/

Agoda | Amazon | Acer | (4196) YouTube | All Bookmarks

Problem List | Run | Submit | Premium

Description | Editorial | Solutions | Submissions

Status	Language	Runtime	Memory	Notes
1 Accepted 6 minutes ago	C++	16 ms	13.3 MB	

Code

```
1 /**
2  * Definition for singly-linked list.
3  * struct ListNode {
4  *     int val;
5  *     ListNode *next;
6  *     ListNode(int x) : val(x), next(NULL) {}
7  * };
8  */
9 class Solution {
10 public:
11     ListNode *detectCycle(ListNode *head) {
12         unordered_set<ListNode*> seen;
13
14         while (head) {
15             if (seen.find(head) != seen.end()) {
16                 return head;
17             }
18             seen.insert(head);
19         }
20     }
21 }
```

Restored from local | Upgrade to Cloud Saving | Ln 1, Col 1

Testcase | Test Result

Case 1 | Case 2 | Case 3 | +

</> Source

20°C  
Sunny

Search

ENG  
IN

16:53  
14-02-2025